

Franciszek MARECKI

HARMONOGRAMOWANIE PROCESU LAKIEROWANIA
KAROSERII SAMOCHODOWYCH

Streszczenie. W pracy przedstawiono problem harmonogramowania procesu lakierowania karoserii samochodowych z podziałem oraz bez podziału zadań. Uwzględniono wpływ jakości procesu na harmonogramowanie. Do rozwiązania problemu wykorzystano algorytm podziału i ograniczeń.

1. WPROWADZENIE

Problem sterowania procesem lakierowania karoserii samochodowych stanowił przedmiot niewielu prac. Modele procesu lakierowania karoserii (samochodu FIAT-126P) przedstawiono w [2, 9]. Złożoność tych modeli wynika z dużej liczby ograniczeń oraz zakłóceń losowych procesu.

Istota problemu sterowania procesem lakierowania karoserii polega na określeniu kolejności (krótkoterminowych harmonogramów) przepływu partii karoserii przez dwa szeregowo agregaty (spawalnię i tunel lakierni). Rozwiązanie tak ogólnie sformułowanego problemu przedstawiono w [6,7]. W pracach tych dopuszczono dowolne uporządkowanie zadań w magazynie buforowym pomiędzy agregatami.

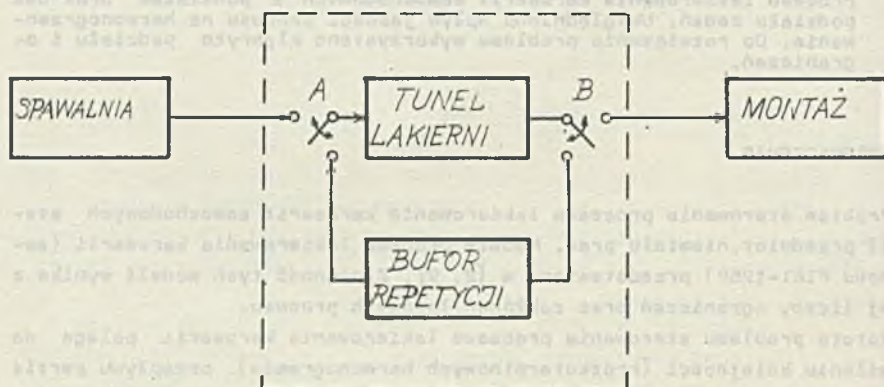
W pracy [3] proces lakierowania karoserii przedstawiono bardziej szczegółowo, wyróżniając tzw. wersje i odmiany kolorystyczne karoserii. Ponadto założono, że kolejność zadań w spawalni jest ustalona, natomiast w magazynie buforowym (między agregatami) istnieją ograniczenia przepływu zadań. Modele dyskutowane w [3, 6, 7] mają charakter deterministyczny.

W procesie lakierowania karoserii istotne znaczenie mają zakłócenia losowe. W wyniku tych zakłóceń pewne karoserie są lakierowane powtórnie. Takie przypadki rozpatrywano w pracach [4, 8]. W niniejszej pracy przedstawiono modele i algorytmy harmonogramowania w pewnych przypadkach zakłóceń procesu. W szczególności rozpatrywane będą modele bez podziału i z podziałem zadań. Do rozwiązania sformułowanych problemów wykorzystano koncepcje algorytmu podziału i ograniczeń zaczerpnięte z [1, 5].

Z teoretycznego punktu widzenia rozważany będzie problem harmonogramowania zadań na jednym agregacie z repetycjami. Repetycje polegają na powtarznej realizacji zadań na agregacie.

2. SFORMUŁOWANIE PROBLEMU

Rozważmy proces lakierowania karoserii samochodowych pokazany na rysunku 1. Zasadniczym agregatem w tym procesie jest tunel lakierni. Przez tunel przepuszcza się karoserie, które znajdują się na transporterze podwieszonym. Transporter ten porusza się ze stałą prędkością. W czasie pobytu karoserii w tunelu lakierni zostaje na nie nałożona warstwa lakieru. Dla zachowania wymagań jakościowych warstwa ta musi mieć określoną grubość. Grubość warstwy lakieru jest zależna od czasu pobytu karoserii w tunelu, a więc od prędkości transportera.



Rys. 1. Model lakierni karoserii samochodowych

A - wejście do tunelu dla karoserii ze spawalni lub z magazynu buforowego dla repetycji, B - wyjście z tunelu do montażu lub magazynu buforowego dla repetycji

Charakterystyka jakości procesu lakierowania określa udział procentowy karoserii dobrych (lub złych) w zależności od prędkości transportera. Charakterystyki takie są dla różnych lakierów i kolorów - różne.

W procesie lakierowania karoserii samochodowych występuje typowy dylemat związany z jakością i ilością produkcji. Przy odpowiednio małej prędkości transportera można uzyskać wyłącznie dobre karoserie. Jednakże w tym przypadku liczba karoserii uzyskanych w pewnym przedziale czasu jest mała. Przy większej prędkości transportera zwiększa się liczba otrzymanych karoserii, lecz wśród nich są karoserie złe (nie spełniające wymagań jakościowych). W związku z powyższym dla harmonogramowania procesu lakierowania karoserii należy określić prędkość transportera.

W procesie lakierowania karoserii - przy określonej prędkości transportera - istotne znaczenie ma kolejność kolorów. Zmiana koloru w tunelu lakierni jest związana z "czasem martwym", w którym atmosfera tunelu jest oczyszczona z resztek emulsji. Czas martwy jest zależny od poprzedniego i

następnego koloru w tunelu lakierni. Przy odpowiedniej kolejności kolorów można zminimalizować sumaryczny czas martwy. Dla harmonogramowania procesu lakierowania karoserii należy określić kolejność kolorów w tunelu lakierni.

Kolejność kolorów nie jest dowolna. Karoserie są przekazywane do lakierni ze spawalni. Z uwagi na charakter procesu spawalniczo-montażowego karoserie są wytwarzane partiami o określonej wersji. Wersja jest określona poprzez kształt karoserii. W ramach wersji wyróżniamy partie karoserii lakierowane na odmienne kolory. Karoserie różnych wersji mogą być lakierowane na ten sam kolor. Z tego względu zadaniem jest (dla tunelu lakierni) pomalowanie partii karoserii określonej wersji na określony kolor.

Realizacja zadań jest ograniczona w czasie. Dla każdego zadania określony jest termin najwcześniejszego rozpoczęcia jego realizacji. Termin ten wynika z dostępności zadania ze spawalni oraz dostępności lakieru określonego koloru (w tunelu lakierni). Poza tym określone są terminy najpóźniejszego zakończenia realizacji zadań. Opóźnienie realizacji zadań powoduje straty. Jako kryterium optymalizacji harmonogramowania procesu lakierowania karoserii samochodowych przyjmujemy minimalizację sumarycznych strat, wynikających z opóźnienia realizacji zadań.

W aktualnej praktyce produkcyjnej stosuje się uproszczone modele procesu. Przyjmuje się zasadę łączenia partii karoserii kolejnych wersji, jeżeli mają ten sam kolor. Ponadto przyjmuje się jednakową charakterystykę jakości procesu dla wszystkich zadań. Przyjmuje się również stałe (niezależne od kolorów) czasy martwe. Prędkość transportera w tunelu lakierni jest dobierana tak, by straty z opóźnionej realizacji zadań były minimalne. Karoserie złej jakości są lakierowane powtórnie lub przekazywane na stanowisko dla usuwania usterek. Dla realizacji zadań wykorzystuje się pewien zapas karoserii w magazynie lakierni. Spełnienie wymagań lakierni jest osiągnięte poprzez powtórne lakierowanie (w tym również lakierowanie na inny kolor). Złożoność i losowość procesu stwarza poważne trudności przy próbie spełnienia wymagań lakierni.

W niniejszej pracy zostaną rozważone dwa odmienne modele harmonogramowania procesu lakierowania karoserii samochodowych. W pierwszym modelu zakładamy, że każde zadanie jest realizowane w całości. Karoserie złe są przekazywane do powtórnego lakierowania, łącznie z niepomalowanymi karoseriami danej partii. Drugi model dopuszcza podział zadań na dwie części. Część pierwszą stanowią karoserie polakierowane dobrze, natomiast część drugą - karoserie złe, przeznaczone do repetycji. Części te nie muszą być realizowane kolejno po sobie. W obydwu modelach zakładamy, że karoseria może być lakierowana co najwyżej dwa razy. Powtórne lakierowania odbywa się na ten sam kolor. Przyjmujemy, że karoserie lakierowane dwukrotnie są dobre, ponieważ dwie warstwy lakieru dają powłokę o wystarczającej grubości.

3. MODEL BEZ PODZIAŁU ZADAŃ

Założmy, że dany jest pojedynczy agregat (tunel lakierni). W agregacie tym są realizowane zadania ze zbioru:

$$\Omega = \{\omega_n\} \quad n = 1, \dots, N, \quad (1)$$

gdzie:

- ω_n - n-te zadanie,
- N - liczba zadań.

Zadanie interpretujemy jako partię karoserii określonej wersji, które są być lakierowane na określony kolor.

Niechaj będą dane chwile najwcześniejszego rozpoczęcia realizacji zadań:

$$\Phi = [\varphi_n] \quad n = 1, \dots, N, \quad (2)$$

gdzie:

- φ_n - chwila najwcześniejszego rozpoczęcia realizacji ω_n .

Analogicznie zapiszemy dane chwile najpóźniejszego zakończenia realizacji zadań:

$$\Psi = [\psi_n] \quad (3)$$

gdzie:

- ψ_n - chwila najpóźniejszego zakończenia realizacji ω_n .

Terminy (2) i (3) są podane od $t=0$. Chwila φ_n nie może być przekroczona - natomiast chwila ψ_n może być przekroczona, lecz jest to związane z stratami (karami umownymi). Dano są jednostkowe kary wynikające z opóźnienia realizacji zadań:

$$S = [s_n], \quad (4)$$

gdzie:

- s_n - kara za opóźnienie ω_n o jednostkę czasu.

Zakładamy, że straty są naliczane dla zadania (całej partii karoserii) od chwili jego zakończenia.

Założmy, że dany jest czas pobytu każdej karoserii w agregacie (tunelu lakierni). Czas ten jest zależny od prędkości transportera - oznaczymy go przez $d(v)$.

Ponadto założymy, że dane są czasy realizacji zadań:

$$\Theta = [\theta_n], \quad (5)$$

gdzie: θ_n - czas realizacji ω_n .

Ponieważ zadanie jest związane z partią karoserii zawieszonych sekwencyjnie na transporterze, zatem ω_n określa czas wchodzenia partii do tunelu lakierni (lub inaczej czas wychodzenia partii karoserii z tunelu lakierni).

Załóżmy, że dane są charakterystyki jakości lakierowania karoserii. Przedstawiają one zależność udziału karoserii złych w lakierowanej partii od prędkości transportera:

$$U(v) = [u_n(v)] \quad (6)$$

gdzie:

u_n - udział karoserii złych w partii ω_n ,
 v - prędkość transportera,

przy tym:

$$0 \leq u_n(v) \leq 1 \quad (6a)$$

Załóżmy, że dany jest czas δ potrzebny na przetransportowanie karoserii przeznaczonej do powtórnego lakierowania. Karoseria taka jest przeznaczona z punktu kontrolnego do tunelu lakierni na transporterze podwieszonym. Transporter ten stanowi magazyn buforowy dla repetycji.

W modelu bez podziału zadań założymy, że:

$$\bigvee_{1 \leq n \leq N} \omega_n u_n(v) \geq d(v) + \delta \quad (7)$$

Z warunku tego wynika, że podział zadania nie jest konieczny.

Załóżmy, że dane są czasy martwe, związane ze zmianą koloru w tunelu lakierni:

$$T = [\tau_{k,n}] \quad (8)$$

$k=1, \dots, N,$
 $n=1, \dots, N,$

gdzie:

$\tau_{k,n}$ - czas martwy pomiędzy realizacją zadania ω_k oraz zadania ω_n .

Czasy martwe wynikają z faktu, że niektóre zawieszki wchodzące do tunelu lakierni muszą być puste.

Kolejność realizacji zadań nie jest dowolna. Ponieważ karoserie są przekazywane ze spawalni do lakierni partiami o określonej wersji, zatem każde zadanie należące do wcześniejszej wersji musi poprzedzać zadania należące do późniejszej wersji. Ograniczenia kolejnościowe zapiszemy w postaci macierzy:

$$\Gamma = [\gamma_{\phi,n}] \quad (9)$$

$\phi=1, \dots, N$
 $n=1, \dots, N$

Elementy tej macierzy mają następujące znaczenia:

$$\delta_{\varphi,n} = \begin{cases} 1: & \text{jeśli } \omega_{\varphi} \text{ jest bezpośrednim poprzednikiem } \omega_n. \\ 0: & \text{w przypadku przeciwnym} \end{cases}$$

Zauważmy, że w przypadku gdy zadania nie są dzielone, to część zadania ($u_n(v)$) jest realizowana powtórnie bez opóźnienia. Zatem całkowity czas realizacji zadania ω_n wynosi:

$$\psi_n^* = (1 + u_n(v))\tau_n^* \quad (10)$$

Harmonogram realizacji zadań winien mieć postać:

$$H = \langle \langle \varphi_1, t_1 \rangle \dots \langle \varphi_n, t_n \rangle \dots \langle \varphi_N, t_N \rangle \rangle \quad (11)$$

gdzie:

φ_n - termin wejścia zadania ω_n do tunelu,

t_n - termin wyjścia zadania ω_n z tunelu,

$\langle \cdot \rangle$ - zbiór uporządkowany liniowo.

Z uwagi na ograniczenia kolejnościowe harmonogram dopuszczalny winien spełniać warunek:

$$\forall_{\varphi} \forall_n (\delta_{\varphi,n} = 1) \Rightarrow (t_{\varphi} - d(v) + \tau_{\varphi,n} \leq \varphi_n) \quad (12)$$

Jako kryterium optymalizacji harmonogramowania przyjmujemy minimalizację strat wynikających z opóźnienia realizacji zadań. A zatem funkcja celu ma postać:

$$Q = \sum_{n \in \alpha} (t_n - \psi_n) a_n \rightarrow \min, \quad (13)$$

przy tym:

$$\forall_n (t_n > \psi_n) \Rightarrow (n \in \alpha). \quad (13a)$$

Rozwiązanie tak sformułowanego problemu przedstawimy za pomocą algorytmu podziału i ograniczeń - z regułą wyboru FIFO. W algorytmie takim należy zdefiniować: stan procesu decyzyjnego, wartość etanu, procedurę generowania stanów oraz procedurę eliminowania stanów nieperspektywicznych [5]. Idea algorytmu (z regułą wyboru FIFO) polega na etapowym generowaniu stanów. Każdy stan ostatniego etapu przedstawia dopuszczalny harmonogram realizacji zadań. Harmonogram optymalny otrzymujemy ze stanu o minimalnej wartości.

3.1. Algorytm

W algorytmie dla modelu bez podziału zadań, stan procesu decyzyjnego zdefiniujemy następująco:

Def. 1: Stan $P^{\lambda, \eta}$ jest wektorem o postaci:

$$P^{\lambda, \eta} = \begin{bmatrix} p_n^{\lambda, \eta} \end{bmatrix} \quad (14)$$

$$\begin{aligned} n &= 1, \dots, N, \\ \eta &= 0, \dots, N, \\ \lambda &= 1, \dots, L_{\eta}, \end{aligned}$$

gdzie:

η - numer etapu decyzyjnego,

λ - numer stanu w ramach etapu decyzyjnego,

L_{η} - liczba stanów η -go etapu decyzyjnego.

Elementy wektora (14) mają następujące znaczenie:

$$p_n^{\lambda, \eta} = \begin{cases} t_n: & \text{jeśli zadanie } \omega_n \text{ zostało zakończone} \\ 0: & \text{w przypadku przeciwnym} \end{cases}$$

A zatem stan początkowy $P^{1,0}$ jest wektorem zerowym, natomiast każdy stan N -tego etapu decyzyjnego daje dopuszczalny harmonogram realizacji zadań. Tak więc ze stanu $P^{\lambda, N}$ otrzymamy:

$$t_n = p_n^{\lambda, N} \quad (15a)$$

oraz

$$g_n = t_n - (1 + u_n(v))\phi_n - d(v) \quad (15b)$$

Z każdym stanem związany jego wartość, którą oznaczymy przez $V^{\lambda, \eta}$ i zdefiniujemy następująco:

Def. 2: Wartość stanu jest liczbą, którą wyznaczamy z formuły:

$$V^{\lambda, \eta} = \sum_{n \in \alpha^{\lambda, \eta}} (p_n^{\lambda, \eta} - \psi_n) g_n \quad (16)$$

przy tym:

$$\forall_n (p_n^{\lambda, \eta} > \psi_n) \Rightarrow (n \in \alpha^{\lambda, \eta}) \quad (16a)$$

Tak więc wartość stanu $V^{\lambda, \eta}$ wyraża straty związane z realizacją zadań wg harmonogramu określonego przez stan $P^{\lambda, \eta}$. W obliczeniach zamiast definicji (16) można posługiwać się formułą rekurencyjną:

$$v^{\lambda, \varphi} = \begin{cases} v^{1, \varphi-1}: & \text{jeśli } t_n < \psi_n \\ v^{1, \varphi-1} + (t_n - \psi_n) s_n: & \text{w przypadku przeciwnym.} \end{cases} \quad (17)$$

W formule (17) przyjmujemy:

$$v^{1,0} = 0 \quad (17a)$$

Optymalny stan końcowy wyznaczamy z warunku:

$$\left(\min_{1 \leq \lambda \leq L_N} v^{\lambda, N} = v^{\lambda^0, N} \right) \quad (p^{\lambda^0, N} = p^0) \quad (18)$$

gdzie: p^0 - optymalny stan końcowy.

Jak widać definicja (16) i warunek (18) korespondują z przyjętym kryterium (13) optymalizacji harmonogramowania. Na podstawie optymalnego stanu końcowego (18) wyznaczamy - zgodnie z: (15a) i (15b) - optymalny harmonogram realizacji zadań.

Procedura generowania stanów polega na uzupełnieniu stanu $p^{1, \varphi-1}$ o dopuszczalne zadanie ω_n tak, by w konsekwencji otrzymać nowy stan $p^{1, \varphi}$. Generowanie stanów rozpoczynamy od danego stanu początkowego $p^{1,0}$ i kończymy stanem etapu N-tego. Stany generujemy etapami, tzn. na podstawie stanów $\varphi-1$ -go etapu generujemy stany etapu φ -tego. Na każdym etapie podejmujemy decyzję o realizacji jednego zadania (różnego dla przejścia od stanu $p^{1, \varphi-1}$ do stanu $p^{1, \varphi}$). Ogólna procedura generowania stanów ma następującą postać:

$$\bigvee_n \bigvee_{\varphi} (p_n^{1, \varphi-1} = 0) \wedge (\psi_n \leq T^{1, \varphi-1} + \tau_{k,n}) \wedge$$

$$\wedge [(x_{\varphi,n} = 1) \Rightarrow (p_{\varphi}^{1, \varphi-1} > 0)] \Rightarrow (p^{\lambda, \varphi} = p^{1, \varphi-1} + \Delta P) \quad (19)$$

Elementy wektora ΔP mają następujące wartości:

$$\Delta p_i = \begin{cases} t_n: & \text{dla } i=n \\ 0: & \text{w przypadku przeciwnym} \end{cases} \quad (20)$$

W formule (19) przez $T^{1, \varphi-1}$ oznaczamy chwilę zakończenia realizacji zadania w stanie $p^{1, \varphi-1}$, tak więc:

$$T^{1, \varphi-1} = \max_{1 \leq i \leq N} p_i^{1, \varphi-1} \quad (21)$$

Ostatnie zadanie ω_k zrealizowane w stanie $P^{1, \tau-1}$ wyznaczamy z warunku:

$$T^{1, \tau-1} = p_k^{1, -1} \quad (22)$$

Chwilę t_n zakończenia realizacji przydzielonego zadania ω_n wyznaczamy z formuły:

$$t_n = T^{1, \tau-1} + \tau_{k,n} + (1 + u_n(v)) \delta_n^1 \quad (23)$$

W formułach (23) i (15b) uwidacznia się wpływ wymagań jakościowych na harmonogramowanie procesu lakierowania karoserii samochodowych. Przedział czasu $\langle \tau_n, t_n \rangle$ wyznaczony dla realizacji zadania ω_n jest zależny od: udziału karoserii złych w partii, czasu pobytu zadań w tunelu oraz prędkości transportera. Ponadto należy zauważyć, że czas martwy $\tau_{k,n}$ jest zaokrąglany w górę do wielokrotności cyklu, a zatem jest zależny od v .

Procedura eliminacji stanów nieperspektywicznych polega na pominięciu w obliczeniach tych stanów, które nie prowadzą do rozwiązania optymalnego. W ten sposób zmniejsza się zajętość pamięci komputerowej w obliczeniach - a ponadto może ulec skróceniu czas obliczeń. Czas obliczeń (optymalnych) jest zależny od efektywności procedury eliminowania stanów. Jeżeli procedura eliminuje stan, to nie jest generowana wiązka trajektorii wychodząca z tego stanu, co skraca czas obliczeń. Jednakże stosowanie samej procedury eliminowania stanów wymaga również pewnego czasu. Jeżeli zastosowanie tej procedury nie powoduje wyeliminowania stanu, to czas obliczeń algorytmu rośnie. A zatem czas obliczeń zależy od liczby testowanych i wyeliminowanych stanów.

Niżej przedstawimy eliminację stanów za pomocą reguły dominacji, która jest oparta na następującym twierdzeniu:

Tw. 1: Stan $P^{1, \tau}$ dominuje nad stanem $P^{2, \tau}$, jeżeli jest spełniony warunek:

$$\forall_n \left[(p_n^{1, \tau} > 0) \iff (p_n^{2, \tau} > 0) \right] \wedge (k^{1, \tau} = k^{2, \tau}) \\ (T^{1, \tau} = T^{2, \tau}) \wedge (v^{1, \tau} < v^{2, \tau}), \quad (24)$$

gdzie: $k^{1, \tau}$ - numer ostatniego zadania zrealizowanego w stanie $P^{1, \tau}$.

Dowód tego twierdzenia opieramy na następującej definicji dominacji stanów;

Def. 3: Stan $P^{1, \tau}$ dominuje nad stanem $P^{2, \tau}$, jeżeli spełnia warunek:

$$(p^{1, \tau} \ll p^{2, \tau}) \iff (v^{1, N} < v^{2, N}), \quad (25)$$

gdzie:

\ll - symbol dominacji stanów,

$\lambda_{1,N}^0$ - optymalny stan końcowy otrzymany ze stanu $\lambda_{1,T}$.

Założmy dla dowodu twierdzenia, że ze stanu $\lambda_{2,T}$ realizujemy optymalny harmonogram, który pozwala osiągnąć optymalny stan końcowy $\lambda_{2,N}^0$: Jeżeli spełniony jest warunek (24), to harmonogram ten jest realizowalny również od stanu $\lambda_{1,T}$. Ponieważ równocześnie $v_{1,T} < v_{2,T}$, zatem trajektoria od stanu $\lambda_{1,T}$ pozwala osiągnąć lepszą wartość końcową: $v_{1,N}^* < v_{2,N}^0$.

Ponadto:

$$v_{1,N}^0 \leq v_{1,N}^* < v_{2,N}^0 \quad (26)$$

A zatem, zgodnie z (24) i (26) stan $\lambda_{1,T}$ dominuje nad stanem $\lambda_{2,T}$.

3.2. Dyskusja

W dyskusji do twierdzenia 1 zauważymy, że spełnienie warunku

$$T_{\lambda_{1,T}} = T_{\lambda_{2,T}} \quad (27)$$

jest raczej nieczęste. Z tego względu prawdopodobieństwo spełnienia reguły dominacji jest małe, a stosowanie tej reguły raczej wydłuża czas obliczeń.

Wprowadzenie do (24) warunku:

$$T_{\lambda_{1,T}} \leq T_{\lambda_{2,T}} \quad (27a)$$

pozwała częściej dominować stany. Jednakże z uwagi na założenie ciągłości pracy lakierni, przy:

$$T_{\lambda_{1,T}} < T_{\lambda_{2,T}} \quad (27b)$$

stan $\lambda_{1,T}$ może nie dać dopuszczalnego harmonogramu. Wynika to z braku dostępnego zadania do realizacji w chwili zwolnienia tunelu lakierni. Warunek:

$$T_{\lambda_{1,T}} > T_{\lambda_{2,T}} \quad (27c)$$

nie może być przyjęty w (24), ponieważ wówczas optymalny harmonogram ze stanu $P^{1,2}$ może nie być realizowalny od stanu $P^{1,2}$.

Jeżeli w modelu bez podziału zadań dopuszczamy możliwość przestoju w pracy lakierni (poza czasem martwym), to:

- procedura generowania stanów ma postać:

$$\bigvee_n \bigvee_{\varphi} (p_n^{1,2-1} = 0) \wedge [(\gamma_{\varphi,n} = 1) \Rightarrow (p_{\varphi}^{1,2-1} > 0)] \Rightarrow \\ = (P^{1,2} = P^{1,2-1} + \Delta P) \quad (28a)$$

- chwilę t_n wyznaczamy z formuły:

$$t_n = \max(\varphi_n; \tau^{1,2-1} + \gamma_{k,n}) + (1 + u_n(v)) \varphi_n^k \quad (28b)$$

Ponadto w twierdzeniu 1 można podstawić warunek (27a), przez co reguła dominacji stanów jest efektywniejsza.

4. MODEL Z PODZIAŁEM ZADAŃ

Jeżeli w procesie lakierowania karoserii warunek (7) nie jest spełniony, to zadania muszą być podzielone. Podział zadania oznacza, że pomiędzy pierwszym i drugim lakierowaniem karoserii następuje przerwa czasowa. W przerwie tej mogą być lakierowane karoserie innego zadania.

Wyróżnimy dwa przypadki: dopuszczalnych i niedopuszczalnych przestoju agregatu (tunelu lakierni). Założymy, że jeżeli przestoje agregatu są dopuszczalne, to repetycje następują bezpośrednio po pierwszym lakierowaniu. Przez repetycję rozumiemy powtórne lakierowanie złych karoserii należących do rozważanego zadania. A zatem w tym przypadku zadania są realizowane sekwencyjnie. W przypadku przeciwnym powiemy, że zadania są realizowane kombinacyjnie.

W modelu z podziałem zadań istotne znaczenie ma pojemność magazynu buforowego dla repetycji oraz liczba karoserii dla każdego zadania. Założymy, że dany jest wektor:

$$W = [w_n] \quad (29) \\ n=0, \dots, N,$$

gdzie:

w_0 - pojemność bufora dla repetycji,

w_n - liczba karoserii zadania ω_n .

W buforze dla repetycji przyjmujemy regulamin FIFO. Zgodnie z tym regulaminem karoserie przeznaczone do repetycji wchodzące do magazynu buforowego jako pierwsze pierwsze są dostarczane do tunelu lakierni. Wynika to z faktu, że bufor jest zlokalizowany na transporterze podwieszonym.

4.1. Sekwencyjna realizacja zadań

Przeanalizujemy przypadek sekwencyjnych realizacji zadań, które są dzielone, jeżeli warunek (7) nie jest spełniony. Czas realizacji zadania zawiera wówczas dodatkowy przestój tunelu. Jeżeli dla zadania w_n lakierowanych jest w_n karoserii, to czas realizacji tego zadania wynosi:

$$\tau_n^{**} = \max \left[(1 + u_n(v)) \tau_n^0; d(v) + \delta + \tau_n^0 \right] \quad (30)$$

Uwzględniając (7) w (30) otrzymamy:

$$\tau_n^{**} = d(v) + \delta + \tau_n^0 \quad (31)$$

Przypomnijmy, że dla modelu z podziałem zadań warunek (7) nie jest spełniony (co najmniej dla jednego zadania), tzn.:

$$\exists u_n(v) \tau_n^0 > d(v) + \delta \quad (32)$$

Formuła (30) może być skomentowana następująco. Jeżeli w najgorszym przypadku $u_n(v) \tau_n^0$ karoserii złych wystąpi na końcu danej partii, to czas realizacji zadania:

- wydłuży się o $u_n(v) \tau_n^0$ na skutek repetycji, gdy karoserie przeznaczone do repetycji są dostępne w odpowiednim czasie,
- wydłuży się o $d(v) + \delta$, tzn. o czas potrzebny dla repetycji ostatniej karoserii, jeżeli karoseria ta nie oczekuje w buforze.

Długości z tych dwóch przestojów czasu określa chwilę zakończenia zadania. Jeżeli spełniony jest warunek (7), to przyrost ten wynosi $u_n(v) \tau_n^0$, natomiast dla (32) otrzymamy przyrost równy $d(v) + \delta$. W zakończeniu tej dyskusji odnotujemy, że jeżeli karoserie złe nie występują na końcu partii, to czas realizacji zadania nie ulega zmianie. Wynika to z faktu, że do tunelu lakierni przekazuje się karoserie tego samego zadania, aż do momentu powrotu ostatniej karoserii (jeżeli wymaga repetycji).

W ogólnym przypadku, dla modelu z podziałem zadań, czas realizacji zadania jest losowy. Wynika to z losowego położenia karoserii złych w partii.

W przypadku optymistycznym, tzn. jeżeli:

$$\tau_n^0 > d(v) + \delta, \quad (33)$$

podziału zadania ω_n nie ma. Optymistyczny czas realizacji zadania liczymy wg (10). Przypadek optymistyczny oznacza, że wszystkie karoserie złe występują na początku partii. Jeżeli warunek (33) nie jest spełniony, a karoserie złe występują na początku partii, to optymistyczny czas realizacji zadania ω_n wynosi:

$$\omega_{n,0}^* = \max \left[(1 + u_n(v)) \eta_n^0; u_n(v) \eta_n^0 + d(v) + \delta \right] \quad (34)$$

Przypadek pesymistyczny oznacza, że wszystkie karoserie złe występują na końcu partii. Pesymistyczny czas realizacji zadania ω_n określa formuła (30).

Stosowanie optymistycznego i pesymistycznego czasu realizacji zadania ma uzasadnienie tylko dla modelu z podziałem zadań. W modelu bez podziału zadań położenie karoserii złych w partii nie ma wpływu na czas realizacji zadania.

Jeżeli zadania są realizowane sekwencyjnie (w modelu bez podziału lub z podziałem zadań), to otrzymujemy uogólniony model bez podziału zadań. W modelu tym czas realizacji zadania ω_n wynosi:

$$\tilde{\omega}_n = \begin{cases} \max(1 + u_n(v)) \eta_n^0; u_n(v) \eta_n^0 + d(v) + \delta: \text{optymistycznie} \\ \max(1 + u_n(v)) \eta_n^0; \eta_n^0 + d(v) + \delta: \text{pesymistycznie} \end{cases} \quad (35)$$

Czasy optymistyczny i pesymistyczny są jednakowe, jeżeli spełniony jest warunek (7).

Dla uogólnionego modelu bez podziału zadań można wykorzystać algorytm przedstawiony w punkcie 3.1.

4.2. Kombinacyjna realizacja zadań

Przez kombinacyjną realizację zadań rozumiemy podział zadania na dwie części oraz możliwość wstawienia pomiędzy te części karoserii innego zadania. Algorytm harmonogramowania dla tego przypadku jest nieco inny, a jeżeli przedstawiony w pkt. 3.1.

Stan procesu decyzyjnego definiujemy następująco:

Def. 4: Stan $R^{n,2}$ jest macierzą o postaci:

$$R^{n,2} = [r_{n,j}^{n,2}] \quad (36)$$

$$\begin{aligned} n &= 1, \dots, N \\ j &= 1, 2 \end{aligned}$$

Elementy tej macierzy definiujemy następująco:

$$[r_{n,1}^{\lambda,\varphi}] = \begin{cases} t_{n,1}: & \text{jeśli pierwsza część } \omega_n \text{ została zrealizowana} \\ 0: & \text{w przypadku przeciwnym} \end{cases} \quad (36a)$$

oraz

$$[r_{n,2}^{\lambda,\varphi}] = \begin{cases} t_{n,2}: & \text{jeśli druga część } \omega_n \text{ została zrealizowana} \\ 0: & \text{w przypadku przeciwnym} \end{cases} \quad (36b)$$

Stan początkowy $R^{1,0}$ jest macierzą zerową, natomiast każdy stan końcowy $R^{\lambda,2N}$ jest macierzą o dodatnich elementach. Stan końcowy daje dopuszczalne rozwiązanie problemu. Liczba etapów wynosi $2N$, ponieważ podejmuje się decyzję o realizacji pierwszej i drugiej części każdego zadania.

Def. 5: Wartość $v^{\lambda,\varphi}$ stanu $R^{\lambda,\varphi}$ wyznaczamy z formuły:

$$v^{\lambda,\varphi} = \sum_{n \in \alpha_2^{\lambda,\varphi}} (r_{n,2}^{\lambda,\varphi} - \varphi_n) a_n \quad (37)$$

przy tym:

$$\forall (r_{n,2}^{\lambda,\varphi} > 0) \Rightarrow (n \in \alpha_2^{\lambda,\varphi}) \quad (37a)$$

W obliczeniach wartość stanu wyznacza się rekurencyjnie (analogicznie jak w (17)). Jednakże wartość obliczamy tylko w przypadku, gdy wygenerowany stan powstał z przydzielenia do realizacji drugiej części pewnego zadania.

Generując stany wyróżnimy dwie procedury. Pierwsza procedura dotyczy przypadku pierwszej realizacji partii karoserii, natomiast druga procedura odnosi się do repetycji.

Pierwsza procedura ma postać:

$$\begin{aligned} & \forall_n \forall_{\varphi} (r_{n,1}^{1,\varphi-1} = 0) \wedge [(\delta_{\varphi,n} = 1) \Rightarrow (r_{\varphi,1}^{1,\varphi-1} > 0)] \wedge \\ & \wedge (w_0^{1,\varphi-1} + u_n(v)w_n \leq w_0) \Rightarrow (R^{\lambda,\varphi} = R^{1,\varphi-1} + \Delta R) \end{aligned} \quad (38)$$

Elementy macierzy ΔR :

$$\Delta r_{1,1} = \begin{cases} t_{n,1}: & \text{dla } i = n \\ 0: & \text{w przypadku przeciwnym} \end{cases} \quad (39a)$$

$$\Delta r_{i,2} = 0 \quad (39b)$$

$$i=1, \dots, N$$

Ponadto:

$$w_0^{1,2-1} = \sum_{j \in \alpha_1^{1,2-1}} w_j u_j(v) \quad (40)$$

przy tym:

$$\bigvee_j (r_{j,1}^{1,2-1} > 0) \wedge (r_{j,2}^{1,2-1} = 0) \Rightarrow (j \in \alpha_1^{1,2-1}) \quad (41)$$

Chwilę wyznaczamy z formuły: $t_{n,1}$

$$t_{n,1} = \max(\varphi_n; \tau^{1,2-1} + \zeta_{k,n}) + \psi_n \quad (42)$$

Ostatnia zadanie ω_k (pierwszą lub drugą jego część) zrealizowane w stanie $R^{1,2-1}$ wyznaczamy z warunku:

$$\max_{1 \leq i \leq N} \max_{1 \leq j \leq 2} r_{i,j}^{1,2-1} = \tau^{1,2-1} = \max_{1 \leq k \leq 2} r_{k,j}^{1,2-1} \quad (43)$$

Jak widzimy w procedurze tej istotne znaczenie mają ograniczenia kolejnościowe oraz ograniczenie pojemności magazynu buforowego dla repetycji. Warto zauważyć, że ograniczenia kolejnościowe nie dotyczą repetycji.

Druga procedura generowania stanów jest oparta na przydzielaniu do realizacji karoserii znajdujących się w buforze dla repetycji. Ponieważ w buforze dla repetycji obowiązuje regulamin FIFO, zatem procedura generowania stanów ma postać:

$$\bigvee_n \bigvee_{\emptyset} (r_{n,1}^{1,2-1} > 0) \wedge (r_{n,2}^{1,2-1} = 0) \wedge (\min_{j \in \alpha_1^{1,2-1}} r_{j,1}^{1,2-1} = r_{n,1}^{1,2-1}) \Rightarrow$$

$$\Rightarrow (R^{2,2} = R^{1,2-1} + \Delta R) \quad (44)$$

Elementy macierzy ΔR określamy następująco:

$$\Delta r_{i,1} = 0 \quad (45)$$

$$i=1, \dots, N$$

$$\Delta r_{1,2} = \begin{cases} t_{n,2}; & \text{dla } i = n \\ 0; & \text{w przypadku przeciwnym} \end{cases} \quad (46)$$

przy tym:

$$t_{n,2} = \begin{cases} T^{1,0-1} + \tilde{\tau}_{k,n} + u_n(v) \cdot \tau_n^0 & \text{dla } k \neq n \\ r_{n,1}^{1,0-1} + \max(d(v) + \delta, u_n(v) \cdot \tau_n^0) & \text{dla } k = n \end{cases} \quad (47)$$

Procedura eliminacji stanów przez dominację nie będzie przytaczana, ponieważ jest czasochłonna (nieefektywna).

5. ZAKOŃCZENIE

W pracy przedstawiono modele i algorytmy harmonogramowania procesu lakierowania karoserii samochodowych. Z teoretycznego punktu widzenia rozważano problem harmonogramowania zadań na jednym agregacie z repetycjami. W sensie złożoności obliczeniowej był to problem NP-zupełny, stąd do jego rozwiązania zastosowano algorytm podziału i ograniczeń. W algorytmie tym wykorzystano regułę wyboru FIFO i dominację stanów. Generowanie stanów metodami zgłębiania nie było dyskutowane, ponieważ - jak dotąd - nie określono efektywnej funkcji dolnego ograniczenia (dla eliminacji stanów). Jeżeli w trakcie obliczeń przekroczony zostanie dopuszczalny obszar pamięci, to otrzymane rozwiązanie jest przybliżone. W takim przypadku na podstawie danego zbioru rozwiązań dopuszczalnych można oszacować prawdopodobieństwo, że w zbiorze tym znajduje się rozwiązanie optymalne.

Czas obliczeń w przedstawionych problemach zależy od struktury danych. Istotne znaczenie ma przy tym zarówno konstrukcja programu, jak i język programowania.

LITERATURA

- [1] Kohler H.W., Steiglitz K.: Przeglądowe i iteracyjne metody obliczeniowe, Teoria szeregowania zadań (red. E.G. Coffman jr), WNT, Warszawa 1980, ss. 241-301.
- [2] Kowalowski H., i inni: Optymalizacja harmonogramowania produkcji w dziale tłoczni, Report z pracy n-b, I.A., Pol. Śl. Gliwice 1981, ss. 310-354.
- [3] Marecki F., Ślesieńska I.: Problemy harmonogramowania procesu lakierowania karoserii, ZN Pol. Śl., seria: Automatyka, z. 64, Gliwice 1982, ss. 83-94.
- [4] Marecki F.: Wpływ wymagań jakościowych na harmonogramowanie procesu lakierowania karoserii samochodowych, 9 Dni Jakości i Niezawodności, NOT, Gliwice 1982, (referat niepublikowany).
- [5] Marecki F.: Discrete Processes Control, 5-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.

- [6] Ślesiańska I., Marecki F.: Algorytm harmonogramowania niezależnych zadań na dwóch szeregowych maszynach, ZN Pol. Śl., seria: Automatyka, (złożono w Redakcji).
- [7] Ślesiańska I., Marecki F.: Algorytm harmonogramowania zależnych zadań na dwóch szeregowych maszynach, ZN Pol. Śl., seria: Automatyka, (złożono w Redakcji).
- [8] Wikierska I.: Uprawieniye procesom okraski kuzow, 5-th International Conference on "Control Systems and Computer Science", Politechnical Institute of Bucharest, Bucharest 1983.
- [9] Zgorzyk M.: Modelowanie cyfrowe lakierni samochodów małolitrażowych FIAT-126, Praca dyplomowa magisterska, I.A., Pol. Śl., Gliwice 1980.

Recenzent: Doc. dr hab. inż. Andrzej Gościński

Wpłynęło do Redakcji: sierpień 1983 r.

РАСПИСАНИЕ ГРАФИКОВ РАБОТ ПРОЦЕССА ОКРАСКИ КУЗОВОВ АВТОМОБИЛЕЙ

Р е з ю м е

В работе представлена проблема составления графиков работ для процесса окраски кузовов автомобилей - с делением и без деления заданий. Учтено влияние качества на расписание графика. Для решения задачи использовано алгоритм раздела и ограничений.

SCHEDULING OF CAR-BODY VARNISH PROCESS

S u m m a r y

In the paper a problem of scheduling of car-body varnish process is formulated. A parameter of the process quality has been included in the model. The problem is solved using branch and bound algorithm.